Project 3 Documentation

Model Description:

Model Components:

1. keras.layers.Conv2D(32, (3, 3), padding='same', activation=tf.nn.relu, input_shape=(32, 32, 3)),
2. keras.layers.Conv2D(64, (3, 3)),
3. keras.layers.MaxPooling2D(2, 2),
4. keras.layers.Dropout(0.2),
5. keras.layers.Conv2D(32, (3, 3), padding='same', activation=tf.nn.relu, input_shape=(32, 32, 3)),
6. keras.layers.MaxPooling2D(2, 2),
7. keras.layers.Dropout(0.1),
8. keras.layers.Conv2D(32, (3, 3), padding='same', activation=tf.nn.relu, input_shape=(32, 32, 3)),
9. keras.layers.Conv2D(64, (3, 3)),
10. keras.layers.MaxPooling2D(2, 2),
11. keras.layers.Dropout(0.3),
12. keras.layers.Flatten(),
13. keras.layers.Dense(1024, activation=tf.nn.relu),
14. keras.layers.Dense(512, activation=tf.nn.relu),
15. keras.layers.Dense(10, activation=tf.nn.softmax)

Model Results:

The test accuracy for my model was around 0.7926999926567078.

The test loss for my model was around 0.7969602546691894.

The model took around 298.4258623123169 seconds to run.

Observations:

I observed that lowering my Dropput rate increased the accuracy.

I also observed that scaling the initial input also improves accuracy.

Having multiple layers and epochs also improves accuracy.

Having too many epochs and a very small batch size causes overfitting.

Having too few epochs and a very large batch size cause underfitting.